USING SAS/AF® SOFTWARE TO BUILD AND TEST END-USERS SAS/AF® SOFTWARE APPLICATIONS

Janet C. Lind, San Diego Data Processing Corporation

ABSTRACT:

One of the goals of the Information Center is to make the software as user-friendly as possible. With the advent of SAS/AF, a user-friendly menu/screen dialog system is now available for end-users. But, what about the person, many times an end-user also, who is chosen to write the SAS/AF application. This end-user writer has to learn the new SAS/AF menu and screen language as well as test it. This paper addresses the streamlining of the SAS/AF screen generation and testing process. It uses a series of SAS/AF menus and screens to drive the building and displaying of end-user SAS/AF menus and screens.

INTRODUCTION:

San Diego Data Processing Corporation uses ISPF Dialog Management System extensively. The Information Center SAS System is driven entirely by ISPF Dialogs. However, it was always a problem when end-users wanted application dialogs because it required one of the techinal support staff to write an ISPF dialog for the end-user. Now, with the advent of SAS/AF, most of this can be turned over to experienced SAS writers, with the involvement of the techical staff limited to consulting about operating systems interfaces, file allocations and final review.

However, when we were testing SAS/AF and writing our first SAS/AF dialogs, it became evident we needed to simplify the mechanics of repetitively invoking PROC BUILD and DISPLAY. This is particularly true in our installa-ation because our end-users have been spoon-fed on ISPF Dialogs and have not been trained in the Display Manager. Our users have no idea the amount of TSO and SAS initialization/setup that goes on behind the ISPF dialogs they use.

The approach we used was to include SAS/AF as one of the options in our ISPF SAS Primary Menu. We next wrote a SAS/AF application to simulate what a user would do under the Display Manager to write and test a SAS/AF software application. This paper will discuss the screens developed, the code behind those screens, general standards, debugging hints, and SAS/AF coding techniques.

SAS PRIMARY OPTION MENU:

The SAS Menu is first presented to the user. SAS/AF was incorporated in the existing ISPF menu with all the other SAS functions. Please note option "A".

```
SDDPC    ----------   SAS  MENU  --  (VERSION 5)  ----------------
SELECT OPTION ===> _
                                                  USERID  - TSOSJCL
* =  New options with Version 5                   PREFIX  - TSOJCL

* A  AF       - SAS/AF screen generation & testing facility
* C  CONVERT  - Convert screen/spread/letter to Ver 5 format
  D  DATA     - SAS data definition and data entry (edit)
  E  EXEC     - SAS program edit (SPF) and execute
  G  GRAPH    - SAS graphs and replay of saved SAS graphs
* L  LETTER   - SAS letter editing and printing
  P  PLAY     - SAS replay of saved SAS graphs
  R  REPORT   - SAS report generation
  S  SPREAD   - SAS spread sheet
  T  TSOUTIL  - TSO utilities only
  U  UTILITY  - SAS utilities - Variables - add/delete/rename/modify
                              Data Sets - copy/display/rename/sort
                              Libraries - copy/ create/reorganize
  Z  ZCOPT    - Copy last panel-generated SAS code
```

SAS/AF TESTING FACILITY:

When the user selects option "A", control is passed to a TSO CLIST which creates a default SAS/AF data library named 'prefix.SAS.AF' (if one does not exist) invokes SAS, and executes the procedure PROC DISPLAY that controls the AF Generation and Test Menu. The code behind these menus and screens is listed in Appendix A.

```
                SAS/AF Testing Facility

1 - Create or modify SAS/AF menus, programs, and help screens

2 - Test SAS/AF menus, programs, and help screens

3 - Display SAS Library catalogs and datasets

4 - Create or modify SAS/AF dialog followed by test Af dialog
        (combination of 1 and 2)

5 - Print AF catalog or copy to a sequential disk file
        (not yet implemented - waiting for SAS 5.??)

6 - Display sample SAS/AF dialogs

Press "ENTER" to execute              Press "PF3" to return
```

Option 1:
Create/modify software application screens. Once the user enters the "AF Library" and "AF Catalog" these names are stored as macros values so they may be displayed on all subsequent screens (to reduce keystrokes). The library specified on the screen is allocated and PROC BUILD is executed. We always allocate the end-user's SAS library to DDNAME/LIBREF "AFDD" so the end-user and the TSO CLIST writer can depend on that DDNAME whenever writing or interfacing with SAS/AF code. The SAS/AF user generally specifies the default SAS library although any existing library can be used.

```
          SAS/AF Dialog Screen Generation
Enter Library and Catalog Containing AF Screens:
AF Library Name ===> _____________________________________
AF Catalog Name ===> _____

Press "PF3" to execute              Enter "CANCEL" to exit
```

Option 2:
Test application menus or screens. The library specified on the screen is allocated and PROC DISPLAY is executed pointing to the selected catalog and menu or screen. The "generated program code" may be listed which is useful when debugging.

```
          SAS/AF Dialog Screen Test
Enter Library, Catalog and Screen to be tested:

AF Library Name      ===> _____________________________
AF Catalog Name      ===> _____
AF Screen Name/Type  ===> _____________  (Ex:  PRIMARY.MENU)
List Code Generated  ===> _              (Yes = any character)
                                         (No    leave blank  )
Press "PF3" to execute                   Enter "CANCEL" to exit
```

Option 3:

Manage catalogs/data sets. The library specified on the screen is allocated and PROC DATASETS is executed. This allows the end-user to perform all of the utility functions without leaving the SAS/AF Testing Facility.

```
          SAS/AF SAS Data Library Display and Maintenance
Enter Library to be displayed:
AF Library Name ===> _____________________________________

Press "PF3" to execute              Enter "CANCEL" to exit
```

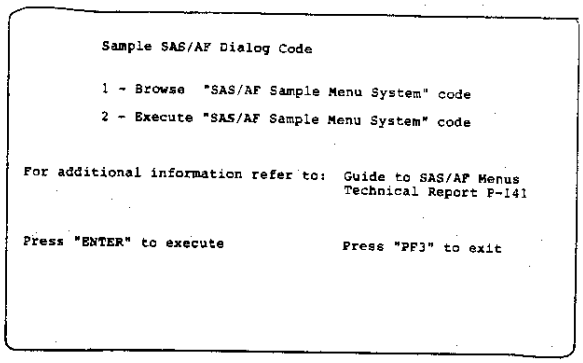
Option 4:
Create or modify SAS/AF software application followed by a test of SAS/AF software application. This is a combination of option 1 and 2. This eliminates the steps of allocating the SAS/AF library a second time and returning to the primary menu to test the application.

```
       SAS/AF Dialog Screen Generation and Test
Enter Library, Catalog and Screen to be tested:

AF Library Name      ===> _____________________________
AF Catalog Name      ===> _____
AF Screen Name/Type  ===> _____________  (Ex:  PRIMARY.MENU)
List Code Generated  ===> _              (Yes = any character)
                                         (No    leave blank  )
Press "PF3" to execute                   Enter "CANCEL" to exit
```

Option 5:
Print the members of a SAS/AF catalog or copy the members to a sequential disk file. This is not yet implemented. We are waiting for the Version 5 Maintenance Release. The option will invoke the new procedure PROC CATOUT to capture all the catalog member names in a SAS Data Set. The SAS Data Set will then be read and SAS code will be generated to print each member. The generated SAS code will be input into a batch job. The output of the batch job is the print file DDNAME "FT12F001". This can be allocated either to SYSOUT or an external disk file. The reason for using an external file is to create a machine readable file of menu and program formats so they can be copied to a application user's manual for documentation. Until the maintenance release, when end-users want a copy of the of the menu and screens they must code their own JCL specifying each member to be printed or copied.

```
   //jobname JOB ...
   //       EXEC SAS
   //FT12F001 DD DSN=TSOiii.print.data,DISP=OLD)
or //FT12F001 DD SYSOUT=...
   //AFDD     DD DSN=TSOiii.SAS.AF,DISP=SHR
   //SYSIN    DD *

       PROC DISPLAY C=AFDD.catalog.name.MENU;
       PROC DISPLAY C=AFDD.catalog.name.PROGRAM;
       PROC DISPLAY ...
       ...
```

Option 6:
Display sample SAS/AF code. This allows a user to browse and execute the "SAS Sample Menu System" distributed with Version 5 (Technical Report: P-141 "Guide to SAS/AF Menus"). The SAS data library is allocated with a DISP of SHR so the users may not modify it. A few examples are worth many pages in a reference manual.



GENERAL STANDARDS

As you gain more experience with SAS/AF you begin to develop lists of "dos" and "don'ts". I'd like to share a few of our "dos".

1. Establish a common DDNAME/LIBREF name to be used when allocating a SAS/AF application library (ex: AFDD). This is no different than SAS's standard names such as SASUSER SASUTL, FT12F001. This forces all internal SAS/AF program code to reference the same DDNAME for branching statements such as PROC DISPLAY AFDD.cat.program.PROGRAM or >>> AFDD.cat.program.PROGRAM. Therefore, any TSO CLIST or SAS code written to invoke SAS/AF can depend on the DDNAME to be used without having to analyze the detail program code.

2. Use a separate catalog for each SAS/AF dialog. This makes maintenance much easier. Select a descriptive name.

3. Establish a common initial menu or program name to start all SAS/AF applications (Ex: PRIMARY.MENU or PRIMARY.PROGRAM). If this is done, then there is no question on how to initiate the software application and you don't have to browse the catalog members to determine the starting place. Note: The descriptive catalog name above provides a clue to the type of application.

4. Establish a SAS/AF library for each application area. Each catalog within the library will contain a specific application dialog. Don't put too many applications in the same library because of the inability to update a SAS Library when other users are allocated to it. Remember, in order to perform emergency maintenance all users of that application library must be exited from that SAS/AF application.

5. Be specific on each menu and screen about the use of the "ENTER" and the "PF3 (END)" key as well as the use of the "CANCEL" command. The users of the application software don't know whether they are executing a menu or a program screen. It is very confusing when the "PF3" key sometimes begins execution and other times returns (exit) to the prior screen.

DEBUGGING HINTS

Debugging SAS/AF can be a real challenge. These are a few techniques we have developed. Remember our installation does not use Display Manager for SAS/AF.

1. Use OPTIONS NOTES; as the first statement in the program code. Set it to OPTIONS NONOTES at the conclusion of testing.

2. Use the PUT statement, if you are in a DATA step, to display the contents of variables.

3. Use the OBS=n on the SET statement to limit the number of observations to be processed.

4. Use the option LIST on the PROC DISPLAY statement to write the SAS code generated by the PROGRAM screen to the SAS log.

5. Test complicated SAS code outside of SAS/AF first. Then copy it into a SAS/AF screen using the INCLUDE command of the text editor.

6. As a last resort, when in the text editor of PROC BUILD, set DMS ON (COMMAND line) in the program screen causing the problem. Then execute PROC DISPLAY under the Display Manager and hopefully a useful message, buried in a multitude of useless messages, will be displayed on the log.

CODING TECHNIQUES

When processing program screens, we have encountered several limitations which we have solved as follows. We chose not to use the macro facility for the time being because of the lack of training and the level of SAS experience of our end-users.

1. The program screen allows you to test only the absence or presence of a field, not the value of the field. We needed an "IF" statement to test the value of fields entered on the program screen and set values of other SAS variables. To solve this, a DATA _NULL_ step was used to test values from a program screen and SYMPUT was used to set macro variables. These macro variables are then available for use in subsequent DATA or PROC steps. Note: When using this technique with character fields of length one, only numeric values will work correctly with the IF statement. (This problem has been referred to SAS Technical Support).

Menu Selection Example from a program screen

```
Select file to be edited:     &

  1 - Applicant File

  2 - Transaction File
```

```
DATA _NULL_;
IF &FIELD1 = 1 THEN DO;
   CALL SYMPUT('MEMBER','APPLFILE');
   CALL SYMPUT('DSN','TSOiii.SAS.APPL');
   END;
IF &FIELD1 = 2 THEN DO;
   CALL SYMPUT('MEMBER','TRANFILE');
   CALL SYMPUT('DSN','TSOiii.SAS.TRAN');
   END;
RUN;                              /*  This RUN is critical */
TSO ALLOC F(SASDATA) DA('&DSN') OLD REUSE;
TSO ALLOC F(SCRDD)   DA('&DSN') SHR REUSE;
PROC FSEDIT DATA=SASDATA.&MEMBER
            SCREEN=SCRDD.SCREEN.&MEMBER..SCREEN;
RUN;                              /*  This RUN is critical */
TSO FREE  F(SASDATA SCRDD);
PROC DISPLAY;                     /*  Return to prior menu */
RUN;
```

2. A date field, defined as numeric with SAS date formats/informats, can be entered on a program screen. However, the contents of the date field is not really a SAS date value. It is a character string. It must be converted to a SAS date value before it can be used in a DATA step to select observations based on a date value or range. This involves the following steps:

   a. Store the program screen field variable in a SAS variable using an assignment statement.

   b. Unstring the SAS variable to a MM, DD, YY variables using the SUBSTR function.

   c. Create a SAS date variable from the MM, DD, YY variables using the MDY function. The reformated date variable is now in a form that can be used in a subsetting IF statement.

Date Field Example

```
Date to be selected ===> &DATE8__   MM/DD/YY
```

```
-------------------------------------------------------------
TSO ALLOC F(SASDATA) DA('CSD5.A82ACCTG.D01') SHR REUSE;
DATA TEMP (DROP DATE8A DATE8AA MM DD YY);
SET  SASDATA.MANDEB;
RETAIN DATE8AA;

IF _N_ = 1 THEN
    DO;
    ********** DATE8 IS MMDDYY8. FORMAT ******************;
    LENGTH DATE8A $8;
    FORMAT DATE8AA MMDDYY8.;
    * &DATE8 is in quotes because of the slashes (mm/dd/yy);
    DATE8A = "&DATE8";
    * SUBSTRING THE DATE INTO MM, DD, YY;
    MM = SUBSTR(DATE8A,1,2);
    DD = SUBSTR(DATE8A,4,2);
    YY = SUBSTR(DATE8A,7,2);
    * CONVERT THE DATE TO A SAS DATE VALUE;
    DATE8AA = MDY(MM,DD,YY);
    ******************************************************;
    END;

IF STRDATE = DATE8AA;
RUN;
TSO  FREE F(SASDATA);
PROC FSBROWSE DATA=TEMP;
RUN;
```

CONCLUSION

SAS/AF provided our corporation with a new tool which could be used to produce application software for our end-users by our end-users. However, we needed to simplify the writing and testing of the SAS/AF application software. Our approach was to write a very simple SAS/AF dialog to provide this function rather than teach our end-users Display Manager. Using this method has enabled our experienced SAS users to develop dialogs much faster since the only new SAS coding they had to learn, in addition to SAS/AF itself, were the TSO ALLOCATE and FREE statements. San Diego Data Processing Corporation's "AF Testing Facility" has resulted in less frustration and increased productivity for the end-users during the SAS/AF software application development cycle.

Janet C. Lind
San Diego Data Processing Corporation
5975 Santa Fe Street
San Diego, CA 92109
(619) 236-5977

APPENDIX A: Menu/Screen Code

Primary Menu Code:

```
1 - Create or modify SAS/AF menus, programs, and help screens

2 - Test SAS/AF menus, programs, and help screens

3 - Display SAS Library catalogs and datasets

4 - Create or modify SAS/AF dialog followed by test Af dialog
        (combination of 1 and 2)

5 - Print AF catalog or copy to a sequential disk file
        (not yet implemented - waiting for SAS 5.??)

6 - Display sample SAS/AF dialogs

Press "ENTER" to execute          Press "PF3" to return

        Attributes for MENU & CBT screens

     Name    ........              Libref   *
     Type    ........              Catalog  *
Option    Key    Name       Type        Libref     Catalog
   1       --    AFBUILD    PROGRAM     *          *
   2       --    AFDISP     PROGRAM     *          *
   3       --    AFDATA     PROGRAM     *          *
   4       --    AFTEST     PROGRAM     *          *
   5       --    AFPRINT    PROGRAM     *          *
   6       --    AFSAMP     MENU        *          *
```

Option 1 Code:

Execute PROC BUILD to generate screens.

```
      SAS/AF Dialog Screen Generation (Option 1)

Enter Library and Catalog Containing AF Screens:

AF Library Name ===> &AFDD_____________________________

AF Catalog Name ===> &AFCAT_


Press "PF3" to execute              Enter "CANCEL" to exit
-------------------------------------------------------------
TSO  ALLOC F(AFDD) DA(&AFDD) OLD REUSE;
PROC BUILD C=AFDD.&AFCAT; RUN;
TSO  FREE  F(AFDD);
PROC DISPLAY; RUN;
```

Option 2 Code:

Execute PROC DISPLAY to test screens.

```
        SAS/AF Dialog Screen Test (Option 2)

Enter Library, Catalog and Screen to be tested:

AF Library Name      ===> &AFDD_____________________

AF Catalog Name      ===> &AFCAT_

AF Screen Name/Type ===> &AFSCR_________ (Ex:  PRIMARY.MENU)

List Code Generated ===> &               (Yes = any character)
                                         (No    leave blank  )

Press "PF3" to execute                   Enter "CANCEL" to exit
-------------------------------------------------------------
TSO  ALLOC   F(AFDD) DA(&AFDD) OLD REUSE;
PROC DISPLAY C=AFDD.&AFCAT.&AFSCR #FIELD4 LIST ## ; RUN;
TSO  FREE    F(AFDD);
* Return to AF Driver menu;
PROC DISPLAY C=AFPROD.AF.AFMENU.MENU; RUN;
```

Option 3 Code:

Execute PROC DATASETS to display/manage catalogs/data sets.

```
        SAS/AF SAS Data Library Display (Option 3)

Enter Library to be displayed:

AF Library Name ===> &AFDD_____________________________


Press "PF3" to execute              Enter "CANCEL" to exit
-------------------------------------------------------------
TSO  ALLOC F(AFDD) DA(&AFDD) OLD REUSE;
PROC DATASETS LIBRARY=AFDD; RUN;
TSO  FREE  F(AFDD);
PROC DISPLAY; RUN;
```

Option 4 Code:

Execute PROC BUILD and DISPLAY to generate and test screens.

```
      SAS/AF Dialog Screen Generation and Test (Option 4)

Enter Library, Catalog and Screen to be tested:

AF Library Name      ===> &AFDD_____________________

AF Catalog Name      ===> &APCAT_

AF Screen Name/Type ===> &APSCR_________ (Ex:  PRIMARY.MENU)

List Code Generated ===> &               (Yes = any character)
                                         (No    leave blank  )

Press "PF3" to execute                   Enter "CANCEL" to exit
-------------------------------------------------------------
TSO  ALLOC   F(AFDD) DA(&AFDD) OLD REUSE;
PROC BUILD   C=AFDD.&AFCAT; RUN;
PROC DISPLAY C=AFDD.&AFCAT.&APSCR #FIELD4 LIST ## ; RUN;
TSO  FREE    F(AFDD);
* Return to AF Driver Menu;
PROC DISPLAY C=APPROD.AF.APMENU.MENU;
RUN;
```

Option 5 Code:

Print or copy existing AF catalogs and screens. Not yet developed. It will use PROC CATOUT (Version 5 maintenance release) and several data steps.

Option 6 Code:

Display sample SAS/AF code. It uses PROC BUILD and DISPLAY.

```
      Sample SAS/AF Dialog Code (Option 6)


   1 - Browse  "SAS/AF Sample Menu System" code

   2 - Execute "SAS/AF Sample Menu System" code

For additional information refer to:  Guide to SAS/AF Menus
                                      Technical Report P-141

Press "ENTER" to execute                  Press "PF3" to exit

        Attributes for MENU & CBT screens

     Name   PRIMARY                 Libref   *
     Type   MENU                    Catalog  *
Option    Key    Name       Type        Libref     Cat
   1       --    AFSAMP1    PROGRAM     *          *
   2       --    AFSAMP2    PROGRAM     *          *
   3       --    --------   --------    *          *
   4       --    --------   --------    *          *

                    AFSAMP1.PROGRAM

-------------------------------------------------------------
TSO ALLOC F(MENU) DA('SAS.AFDEMO') SHR REUSE;
PROC BUILD C=MENU.SASMENU; RUN;
TSO FREE  F(MENU);
PROC DISPLAY; RUN;

                    AFSAMP2.PROGRAM

-------------------------------------------------------------
TSO ALLOC F(MENU) DA('SAS.AFDEMO') SHR REUSE;
PROC DISPLAY C=MENU.SASMENU.PRIMARY.MENU; RUN;
TSO FREE  F(MENU);
PROC DISPLAY; RUN;
```